Software Development Agreement – 10 Legal Issues to Cover

If you're a software developer, you'll want to protect yourself by having a customized software development agreement that's designed specifically for your business.

Why?

If you "borrow" someone else's contract to use with your clients, chances are you're (a) not fully protecting yourself and (b) you're committing copyright infringement because you don't any of the intellectual property rights to use the agreement (e.g., a license from the software lawyer who created it).

So, what should you include in your software development agreement?

The terms and conditions in your contract will vary because your business is unique. However, here are 10 of the most common issues you'll want to address in your agreement.

1. Scope of Work. Your contract will cover both what you're agreeing to develop for your client and specifically exclude work that you will not be doing without additional compensation.

Related Article: Keys to A Successful Mobile App Development Agreement

2. Work Change Orders. Because it's likely that the work involved will be modified during the course of the project, you'll want to have your agreement include a mechanism for change orders by you and the client. These change orders should describe the additional work, cover compensation for the work, and any alterations to the project's milestones and deadlines because of the changes.

3. Subcontracting. If you're outsourcing any of the coding to third parties (e.g., programmers overseas in India or the Philippines), your software development agreement should make it clear that you have the right to do so as well as cover any restrictions on such subcontracting (e.g., confidentiality agreements).

4. Delivery and Testing. Your contract should address what constitutes delivery and the milestones for each deliverable. With a possible exception for cowboy coding, your development process will heavily influence these provisions (e.g., whether or not a prototype will be provided). As part of these provisions, you should address the client's rights to test the deliverables, what constitutes acceptance or rejection of a deliverable, and any of your obligations to fix a deliverable that doesn't satisfy specifications.

5. Payment. How and when will you get paid during the development process? Will the client be required to pay a portion of your fee up front? What milestones trigger subsequent payments? As a developer, it's in your financial interest to front load the payments as much as feasible so that you don't have to pursue the client for collection after the project is completed. On the flip side, the client will want to back end the payments in order to ensure your performance of the work.

6. Intellectual Property Ownership. Who owns the software you develop for the client? You or the client? Are you granting a client a license to use the software? If so, what's the scope of the license? If open source libraries are used as part of the development, that should be addressed to in the agreement so that there are no misunderstandings as to who owns what.

Related Article: Software Development – Who Really Owns the Intellectual Property?

7. Warranties and Disclaimers. What type of warranties will you give the client for the software? For how long? What's the mechanism for the client making a warranty claim? What warranties are you specifically disclaiming (e.g., fitness for a particular purpose)?

8. Competition. Will you be able to sell the software to your client's competitors? Can you compete against the client using the software you've developed? If so, under what restrictions? Will there be a waiting period or a geographic limitation?

9. Confidentiality. How will confidential information be handled? What constitutes "confidential information?" When the project ends, what responsibilities (if any) do you have for this data?

Related Article: Software Developer Nondisclosure Agreement – 5 Key Issues to Cover

10. Software Maintenance and Support. Will you have any support or maintenance obligations for the software? Is there any obligation to upgrade the software (e.g., to be compatible with a new OS)?

Of course, these are just some of the major issues you'll want to address in your software development agreement in order to protect yourself (and avoid unnecessary lawsuits in the process). An experienced software lawyer like Attorney Mike Young can customize the contract to meet your specific needs.

About the Author

Attorney Mike Young has been practicing business and technology law since 1994. And he's been an entrepreneur since 1988.

To get legal help from Attorney Young, click here now or call 214-546-4247 to schedule a phone consultation.